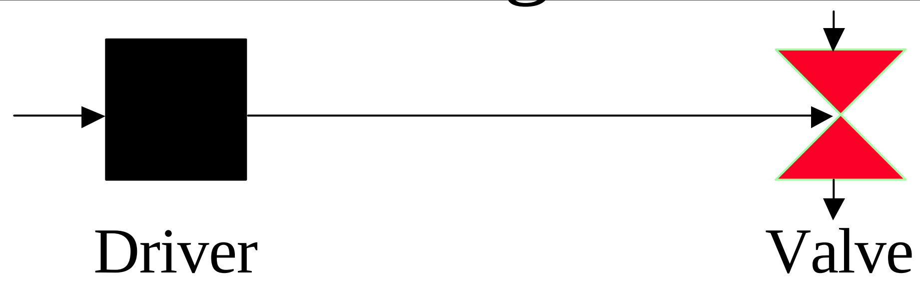
Model-based Execution
and
Model-based Reactive Planning

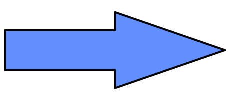
Limitations of Model-based Configuration Management



- All transitions must be directly commandable. Requires hand coding nonlocal control procedures.
  - proc close(valve)
    unless on(driver) turn-on(driver);
    send(driver,close-valve)
- Presumes that commands can be interleaved arbitrarily without destructive interaction or changes in effect.
  - close(valve); turn-off(driver) OK
  - turn-off (driver); close(valve) WRONG

Limitations of Model-based Configuration Management (cont.)

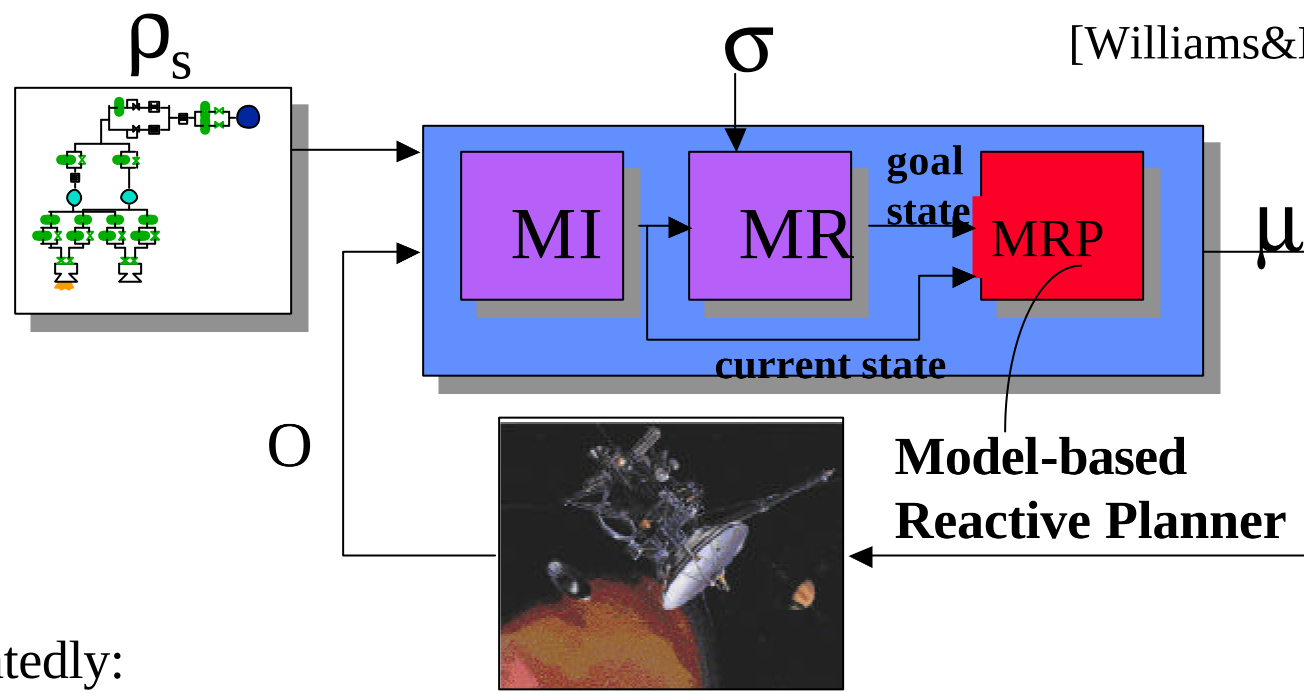
- MR generates a sequence of recovery actions, there is not a tight feedback loop between MR commanding and MI monitoring.
  - MR produces task net
    E.g., close(valve);
    turn-off(driver) where
    proc close(valve)
    unless on(driver) turn-on(driver);
    send(driver,close-valve);



Model-based Reactive Execution

Model-based Reactive Executive

[Williams&Nayak 97]

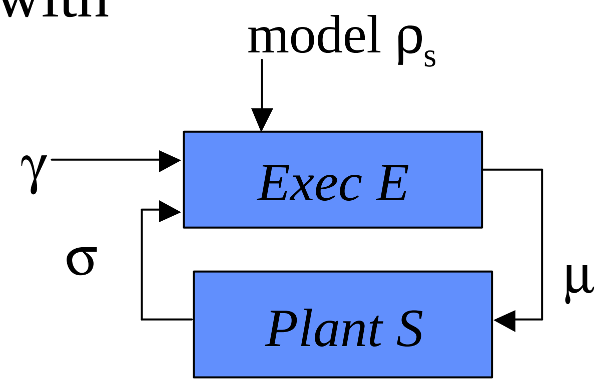


Repeatedly:

- MI generates most likely current state
- MR generates least cost target state
- MRP generates first control action in sequence for reaching the target.
- MI confirms desired effect of first action.

Model-based Reactive Control System $< S, \Theta, E >$

- S is a transition system $< \Pi, \Sigma, T >$ with initial state Θ.
- E is a model-based reactive executive with
  - input goal configurations: $\gamma : g_0, g_1 \ldots$
  - input observables: $\sigma : o_0, o_1 \ldots$
  - output control actions $\mu : \mu_0, \mu_1 \ldots$

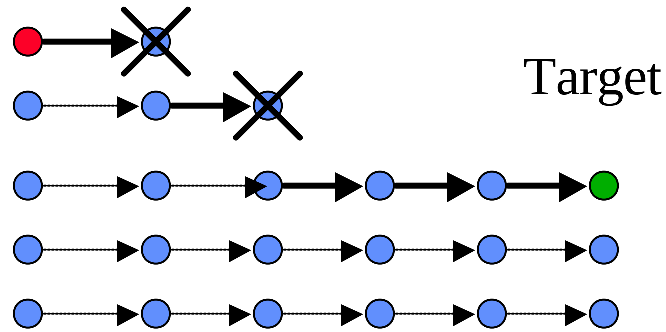


S evolves along trajectory $s : s_0, s_1 \ldots$ s.t:

- $s_0 = \Theta$
- $\sigma_i \in \Sigma$, and agrees with o_i, μ_i and s_i
- $s_{i+1} = \tau(\sigma_i)$
  - results from a failure transition,
  - satisfies g_i or
  - **$<s_i, s_{i+1}>$ is the prefix of a simple (loop free) nominal trajectory that ends in a state satisfying g_i.**

Why plan myopically by assuming the most likely state is correct?

- Reduces computational cost.
- MI gathers information about action affects that confirm or deny this assumption.
- If the most likely state is incorrect, MI moves to a less likely state by elimination
- Eventually the correct state is reached.
- If no irreversible action is taken, this strategy will eventually reach the target, given that it was initially reachable



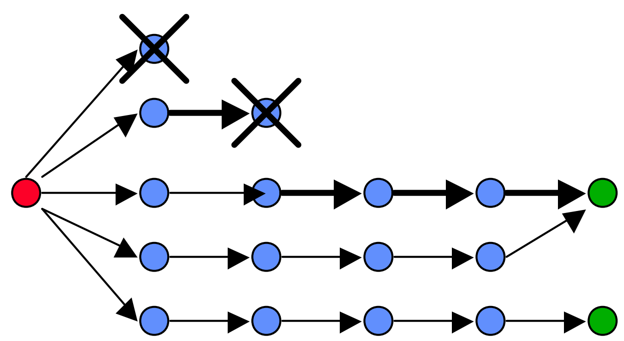
REQUIREMENT 1:

MRP only considers reversible control actions ***except*** *when the only effect is to repair failures.*

How does MR search efficiently over the set of feasible target states?

- How do we enumerate the set of least cost states without generating all possible plans?
- Can the reachable target states be efficiently enumerated by increasing cost?

Target achieves $\mathbf{g_i}$



Solution falls out of the development of the reactive planner